**Course content, Timetable of work and syllabus –** Training content on Front End development course: html/css, JavaScript**:**

* 1. The goal of this document is to describe the main items of the training plan and syllabus on Front end development course based on html, CSS, and JavaScript programming language. Course is based on theoretical discussion through presentations, as well as intensive practical use cases and life coding will be conducted during the training sessions. Final goal is to have two basic iterations: html/CSS classes first to create and develop front end layouts and landing pages; second JavaScript programming language fundamentals and development, fully to maintain the operations on Frond end development.
  2. Full course will take about 6 months, in the end participants must be qualified as a first level Junior engineers and be able to work and develop Front-end application of webpages, platforms and software applications.
  3. **Training plan**: Training classes will be conducted twice each week, 2.5 hours every class to last. In the end of every iteration, examination is planned, based on the evaluation, respective candidates will be able to move to the next stage and continue learning and develop their skills in the JS programming. Introduction to Computer Architecture and Html/CSS course will take totally 40 hours and JavaScript 80 hours, all together will be 120 hours**. Price for an hour is 15,000 drams**
  4. **books for the course**
  5. 1. Code: The Hidden Language of Computer Hardware and Software

1. : Advanced JavaScript Books
2. - David Flanagan, JavaScript. The Definitive Guide
3. - Kyle Simpson, You Don’t Know JS Series
4. - John Resig, Secrets of the JavaScript Ninja Lessons
5. Beginning Programming ALL-IN-ONE DESK REFERENCE FOR DUMmIES

|  |  |  |
| --- | --- | --- |
| **Lesson #** | **Course content / Practice** | **Duration (hours)** |
| 1. Introduction to Operating systems and Technologies in general | * Introduction to PC components, to Operating systems (Windows and Linux), evolution of technologies. * Basic manipulations with OS. (Installing and configuring frameworks. * Introduction to document types, how we define them. | 2 h  2h |
| Introduction to Computer Architecture:  • Definition of computer architecture and its role in computer operation.  • The main components of a computer: processor, RAM, hard drive, video card and motherboard.  2. Central processing unit (CPU):  • The role and functions of the CPU in a computer.  • Processor architecture: blocks, registers, arithmetic logic unit (ALU) and control unit (CU).  • CPU operation: instruction execution, execution loop and interrupts.  3. Random access memory (RAM): • The role and functions of RAM in a computer. • Types of RAM: SRAM (static RAM) and DRAM (dynamic RAM). • RAM organization: memory cells, addressing and data access. | 2h  2h  2h |
| 1. Introduction to Programming | * What is programming * How the code is compiled and how Computer understands * What is Frontend, what is Backend, in which parts project is divided. * What problems can we solve with frontend skills   Introduction to number systems:  • Review of different number systems: binary, decimal, octal and hexadecimal.  • The concept of digit and the meaning of digits in different number systems.  2. Binary number system:  • Definition of the binary number system and its features.  • Representation of numbers in binary system.  • Convert numbers from decimal to binary and vice versa.  3. Operations in the binary system:  • Addition of binary numbers.  • Subtraction of binary numbers.  • Multiplication and division in binary system. | 2h  2h  2h |
|  |  |
| 1. Introduction to HTML | * What is HTML, when it was invented, how it evolved * What is CSS, JSS, in which part of project they are used. * Introduction to tags, their types * Introduction to code editor, what it is and why we use it | 2 h |
| * Making some research, looking at construction of many webpages, web-app and make conclusion that every website construction is made by HTML. * Creating first HTML document. * Real time coding * Learning and use in practice tags such as <p>, <img>, <br>. | 2h |
| 4. Learning new tags | * New tags * Introduction to “Inspect Element”, define that it represents us our last version of HTML, work with it basically | 2 h |
| * Add to HTML document tag such as <span> <a> <h1-h6> * Play with “Inspect Element” * Q&A session |  |
| 5. Introduction to attributes, and styles | * Attributes, examples, and life coding * Introduction to inline styles * How to add videos and photos | 2 h |
| * Real time coding on 0ptional attributes, do exercises with team * Add to texts styles: colour, background-colour, font-size | 2 h |
| 6. Introduction to lists | * Make an example with list (list of products) * How list will be created and show in our page * Difference of creating ordered and unordered list | 2 h |
| * Manipulation with list (creating list in list with different types, add them styles and attributes). * Life coding | 2 h |
| 7. Introduction to embedded Style, <div> | * Styles inside <head>, types of styles, inline-styles * Introduction to <divs>, make point that it is vital tag, * New styles: width, height, background-image; background-size, position, background-repeat, font-weight, font-font-family | 2 h |
| * Practical discussions * Demo and life coding * Q&A | 2 h |
| 8. Introduction to select method: id, class | * Selectors, why we use them, differences * Introduction to <input> tag * Introduction to <button> tag | 2 h |
| * Template making. Do coding, what is margin, how we can use it. | 2 h |
| 9. Introduction to CSS. Forms and tables | * Take our all CSS and add new document and pass all properties to it and connect CSS to HTML * Introduction to property “display”, “float” * What is Forms, show and discuss around examples * Tables and tags: <tbody>, <thead>, <th>, <tr> etc. Tables with properties. Options and Selects. | 2 h |
| * Take our all CSS and add new document and pass all properties to it and connect CSS to HTML. * Learning how suite <div> in one line, and do exercise related to that section | 2 h |
| 10. Forms and tables. | * What is Forms, show them examples. * Speak about tables and learn tags: <tbody>, <thead>, <th>, <tr> etc. * Give tables some property. * Speak about options and selects. | 2 h |
| * Create first form. * Create first table. | 2h |
| 11. Flex, Grid | * Introduction to flex, how we use it * Introduction to gird, tell about its advantages * Define different tags that can be used instead of <div> | 2 h |
| * Create templates with flex * Create templates with the Grid. | 2 h |
| 12-13. | * Practical discussion and life coding * Active learning and demonstrations and coworking with the teams * Coding, testing, development | 2 h |
| * 2 complete lessons continuously basically on life coding | 2 h |
| 14. Animation | * Introduction with Animation * Creating examples | 2 h |
| * Create templates with animation and transition | 2h |
| 15. Responsiveness | * Introduction to responsiveness. * Create Navbar example responsive. | 2 h |
| * Every participant will be given a unique template and makes it responsive. | 2 h |
| 16. Cross Browser Compatibility | * The main goal of this lesson is not to master in Cross Browser template making, it should be during work and practice, we show them that here is technology that you should use and remember during every project. | 2 h |
| * Give example of HTML that in different browsers can be shown differently * Show main ideas and topic about it | 2h |
| **Total** | | **40 h** |

**JavaScript programming language and technologies Training content**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Introduction to JavaScript | * What is JavaScript, what programming language is, how it evolved. * Linking a JavaScript file to HTML. * Values and variables. * Data Types**.** | | 2 h |
| 1. Let, Var, Const. | * Introduction to creating a variable, giving it some value. * Difference of let, var, cont. * Basic Operations. * Operation Precedence. | | 2 h |
| * Code Challenges #1. | |  |
| 1. If/else statement, conditions | * If/else. * Types of Conversion. * Truthy and falsy values. * == vs ===. * Boolean logic. * Logical operations. * Switch statement. * Statement and expressions. * The Ternary operator | | 2 h |
| * Code Challenge #2. * Code Challenge #3. * Code Challenge #4. | | 2h |
| 4. Functions. | | * introduction to functions. * Types of functions, their difference. * Functions calling other functions. * Strict mode. | 2 h |
| * Code challenge #5. |  |
| 5. Arrays. | | * Introduction to arrays. * Array methods. | 2 h |
| * Code Challenge #6. * Code Challenge #7. |  |
| 6. Objects. | | * Introduction to objects. * Object methods. | 2 h |
| * Code Challenge #6. * Code Challenge #7. | 2h |
| 7. Iterations, looping. | | * Introduction to looping. For, while, do-while. * Looping arrays. * Learning new styles: width, height, background-image; background-size, position, background-repeat, font-weight, font-font-family. | 2h |
| * Code Challenge #8. * Code Challenge #9. |  |
| 8. DOM. | | * What is DOM, and DOM manipulation. * Select and manipulate elements. * Click events. * Manipulate CSS with JavaScript | 2 h |
| * Project #1 (Guess my number). * Project #2 (First template with HTML CSS JS) | 2 h |
| 9. JavaScript Engine and Runtime. | | * What is engine. Engine types. * Execution Context, call Stack. | 2 h |
| 10. Scope and scope chain. | | * Introduction to Scope. * Scope Chain * Hoisting. * TDZ. * “This” keyword. * Regular vs. Arrow functions. * Primitives vs. Objects. | 2 h |
| 11. Data Structure. | | * Desctructuring Arrays. * The Spread operator. * Rest pattern. * && and || and ??. * Logical Assignments Operators. * For-of loop. * Define different tags that can be used instead of <div>. | 2h |
| * Code Challenge #10. * Code Challenge #11. | 2 h |
| 12. Looping Objects. | | * Optional Chaining. * Keys, values, entries. * Sets. * Maps. * Maps iteration. | 2 h |
| * Code Challenge #12. * Code Challenge #13. | 2 h |
| 13. Strings. | | * Introduction to String methods. | 2h |
| * Code Challenge #14. |  |
| 14. Functions. | | * How passing arguments work. * High-Order Functions. * Call-back functions. * Functions returning other functions * Call, apply, bind. | 2 h |
| * Code Challenge #15. * Code Challenge #16. | 2h |
| 15. Array method in Projects. | | * Project #3 (Bankist App) * Use array methods in practice. | 2h |
| 16. OOP. | | * Introduction to OOP. * Constructor functions. * Prototypes. | 2 h |
| 17. Classes. | | * ES6 Classes. * Getters and Setters. * Static method. * Encapsulation. | 2 h |
| 18. Project #4 (OOP in practice) | | * Creating OOP-based app with map logic, geolocation (google-maps). | 2 h |
| 19. Async/await. | | * Asynchronous JavaScript, AJAX and APIs. * XMLHttpRequest. * Callback Hell. | 2 h |
| * Code Challenge #17. * Code Challenge #18. | 2 h |
| 20. Promise. | | * Introduction to Promise, advantage of using it. * Consuming and chaining Promise. * Handling reject Promise. * Event Loop. * Promise with async await. * Try catch. * Promise race, any, all | 2 h |
| * Code Challenge #19. * Code Challenge #20. | 2 h |
| 21. Modules. | | * Importing and exporting modules. * Top-level await. * The module patterns. * NPM. * Declarative and Imperative pattern. * Try catch. * Promise race, any, all | 2 h |
| 22. GIT, Gitlab, GitHub. | | * introduction to GitLab. * introduction to GitHub. * Basic git commands. | 2 h |
| * Creating repository in GitHub. * Pulling, adding, committing, pushing, merging. |  |
| **Total** | |  | **80 h** |